

Scalable and Flexible Location-Based Services for Ubiquitous Information Access

Rui José¹ and Nigel Davies²

¹ Information Systems Department, University of Minho,
Azurém, 4800 Guimarães, Portugal
`rui@dsi.uminho.pt`

² Distributed Multimedia Research Group, Department of Computing,
Lancaster University, Bailrigg, Lancaster, LA1 4YR, UK
`nigel@comp.lancs.ac.uk`

Abstract. In mobile distributed environments applications often need to dynamically obtain information that is relevant to their current location. The current design of the Internet does not provide any conceptual models for addressing this issue. As a result, developing a system that requires this functionality becomes a challenging and costly task, leading to individual solutions that only address the requirements of specific application scenarios. In this paper we propose a more generic approach, based on a scalable and flexible concept of location-based services, and an architectural framework to support its application in the Internet environment. We describe a case study in which this architectural framework is used for developing a location-sensitive tourist guide. The realisation of this case study demonstrates the applicability of the framework, as well as the overall concept of location-based services, and highlights some of the issues involved.

1 Introduction

As computation and networking move from the desktop to ubiquitous features of our daily lives, location becomes an increasingly important component of many software systems. In particular, there has been considerable interest in supporting methods of information access in which applications dynamically select the information that is relevant to their current location. Typical examples of localised information are maps, traffic, tourist attractions or shopping places.

The current design of the Internet does not provide any conceptual models for supporting associations between information and location. Existing approaches are based on ad hoc solutions that are only valid for the narrow application domain for which they have been developed or for the specific underlying technology on which they are based [1–3]. The lack of abstractions on which to base the design of such systems makes their development a challenging and costly task, as developers must deal directly with every low-level detail of the system.

* First International Symposium on Handheld and Ubiquitous Computing, HUC'99, Karlsruhe, Germany, September 27-29, 1999. Published by Springer, Lecture Notes in Computer Science Vol. 1707, ISBN 3-540-66550-1.

In this paper, we propose a scalable and flexible concept of location-based services as a generic approach for developing systems that need to associate information with location. In the context of our work, a location-based service is an, otherwise normal, Internet service with a scope that defines its usage in terms of a geographical area. For example, a service providing information on local parking availability may have a scope corresponding to an area of a town, while a service providing directions within a building would have a scope corresponding to that building. Our concept of location-based services is scalable because scopes may range from small locations to wide-area coverage without any significant effect on the systems performance. Furthermore, it is flexible because it is independent of any constraints imposed by specific location or networking technologies and is aimed at the heterogeneous environment of the Internet.

The location-based usage, which characterises the nature of these services, typically results from either some interaction with the physical environment, e.g. controlling the temperature of a room, or from the provision of information associated with a location context, e.g. a local map. In this paper, we focus on this second type of information-centric services and in supporting the association of information with location. Work on the former includes [3–5], and mainly addresses the issues associated with ubiquitous device interaction.

Location-based services represent a simple and yet powerful concept. They are simple because, as Internet services, they can fit in the normal framework of their usage, creation and in many aspects location. They are powerful because they provide a paradigm on which the development of a vast range of location-dependent systems may be based. They allow location dependency to be introduced into an application by adding appropriate service location mechanisms and by defining its behaviour as a function of the services available in its environment. The indirection introduced by the use of service location shields application developers from the complexity of the heterogeneous Internet environment.

This paper describes our model of location-based services, how we propose to support this model, and how it can be used as a generic approach for engineering systems that require localised information. In Section 2, we analyse existing systems, relating them to our work and discussing the reasons why, in our opinion, they are not generic approaches for supporting localised information. In Section 3, we describe the concepts, components and procedures of our architectural framework for location-based services. This is followed, in Section 4, by the description of a case study in which the framework is used for creating a location-sensitive tourist guide. The case study demonstrates the application of the framework and analyses some of the issues involved. Finally, in Section 5, we discuss our future work and present some concluding remarks.

2 Analysis of Existing Approaches

In recent years, a number of systems have addressed the issue of associating information with location contexts. What mainly distinguishes our approach is

the focus on a generic concept, i.e. location-based services, supported by an architecture that is not bound to any specific application scenario or underlying technology.

In its overall objectives, our work has much in common with that of Schilit [6] and Brown [7] in the area of context-sensitive applications, namely we also aim to support systems that react to changes in their environment, and in particular location. Our approach is substantially different in that we build context-awareness from the range of services available in a given location context, whereas both Schilit's and Brown's approaches are based on the use of several types of environmental variables.

The Service Location Protocol (SLP) [8] is the standard proposed by the Internet Engineering Task Force (IETF) for service location within a single administrative domain. As an emerging standard protocol for service location, SLP is an important reference for our work, and we use some of its components. But there are several reasons why we do not consider SLP an adequate technology for the problem of location-based services. Firstly, SLP is not meant to scale to the wide area. It has been created as a dynamic configuration mechanism for applications in local area networks and under a single administrative domain. Secondly, the scope model used by SLP, as well as the corresponding protocol support, are very limited when used for modelling location scopes. In SLP, scopes are only a simple form of administrative service aggregation within a larger implicit scope, which is the administrative domain. They lack hierarchy or any other kind of relationship that could be used for modelling them as locations and they are only valid within the boundaries of their SLP domain.

Wide area service location [9] has been proposed to overcome the limitations of SLP in terms of wide area usage, thus enabling client software to find services in logically remote locations in the Internet. It differs from our work in that the objective is to select a server anywhere on the Internet regardless of location, whereas we use location as the basis for service selection.

Work on addressing and routing based on the Global Positioning System (GPS) [10], which aims to integrate the concept of physical location into the design of the Internet, can also be used to support geographical services. The possibility of geographically routing messages could be used by a server to advertise its services within a certain distance range. Clients could also use this mechanism for finding servers within a certain distance from their current location. This approach differs from our work in its assumption that every node can independently determine its own geographic position and also in its concept of proximity, which is based on the absolute physical proximity between the location of clients and servers.

The work by Imielinski on Wireless Information Systems [2] also explores the idea of location-dependent information services for the mobile wireless environment. This environment is seen as a collection of wireless cells, managed by Mobile Support Stations (MSS). Information pages are accessed from a set of mobile hosts that interact with MSSs to obtain the information they need. The different pages available on each MSS provide the basis for the introduc-

tion of location dependency. Our work may be seen as a generalisation of this approach. Our concept of location-based services may recreate the functionality of this system, while avoiding many of its assumptions such as the existence of a homogeneous wireless system.

The architecture proposed by Hodes in [3], while probably the work that is closest to our notion of location-based services, also presents some differences, both in goals and approach. Its Universal Interaction System is mainly oriented towards device interaction, focusing on the associated aspects of interface adaptation. This is in contrast with our information-centric approach and affects a number of system design decisions, most notably scale. Also, the scope model used in Hodes’s system is closely tied to the network beacons and does not support any form of hierarchy. We propose a scope model that can model physical space and scale to the Internet level.

2.1 Analysis

Overall, existing systems present several types of limitations that restrict their applicability as generic approaches for supporting localised information in the Internet. Essentially, they provide pragmatic approaches to address specific requirements, rather than high level abstractions applicable to a broad range of scenarios. This lack of abstractions leads to systems which are excessively dependent on the specific characteristics of their underlying networking environments, and introduces assumptions that are unique to the specific problem domains being addressed. As a consequence, those approaches are only valid for the narrow and specific application scenarios for which they have been created.

We also consider that current systems do not address the issue of proximity in an adequate way, i.e. as a very abstract concept with significant variations in different contexts and for different activities. A common limitation is only supporting the notion of proximity that results directly from the connectivity range of the specific network technology on which the system is based. We consider it a key requirement that the system is able to support additional layers of proximity on top of the basic notion provided by the underlying technologies.

Another issue concerning proximity is the use of a proximity concept based on the absolute physical distance between servers and clients. This absolute distance does not provide a convenient model for many of our overlapping notions of physical context. For example, if searching for a petrol station while on a motorway a user is interested in the stations located in the service areas of the motorway and not necessarily in the stations in the physical neighbourhood of its current position. This is particularly important for larger scales of space, as the correlation between context and proximity tends to decrease as we enlarge our notion of proximity, i.e. more things that we do not care about will be in our proximity. Using the absolute position of servers has the added disadvantage of forcing servers to be located within the area for which they are providing location-based services. We believe this would be a limiting factor for many application scenarios.

3 Architectural Framework

3.1 Design Goals

Our system is designed to meet two fundamental goals. The first is to provide support for multiple views of space and notions of proximity over the same infrastructure. This is crucial for supporting multiple applications, as they may have very different requirements in how they relate to space and in the proximity range they use. This requirement is addressed by a flexible hierarchy of symbolic locations, which we use as the basis for scope definition.

The second goal is to support mechanisms for locating services that are effectively based on physical location. Such mechanisms should allow two collocated devices, attached to different networks, and under different administrative domains to have access to the same service information. To achieve this goal, we developed a service location architecture based on explicit location scopes and transparent to any network or administrative boundaries.

3.2 Location Scopes

A key component of this architectural framework is a model for representing service scopes in terms of physical space. A basic requirement for such a model is supporting explicit location scopes, i.e. scopes that are not based on any implicit forms of scoping, such as administrative boundaries or network topology. This is essential in guaranteeing that a scope can be shared by multiple applications in the same location, even if they are in different networks or domains.

The essence of our approach is to assume the existence of a shared set of symbolic locations that are used as contexts for service location. The way these locations are supported by the system will be described in the next sections. Servers, when registering their location-based services, use one or more of these location contexts to indicate the scope of their services. Clients, when searching for services, use one of these location contexts to indicate the area on which they wish to find services.

Location contexts are organised in a similar way to the location domain model described in [11]. They are partially ordered by the “contains” relationship, reflecting the spatial inclusion of the associated geographical areas, and they may overlap, forming a lattice instead of a tree, as exemplified in Fig. 1.

Our model for location scopes also includes the possibility of classifying location contexts according to their type, e.g. building, room, or town, allowing applications to know the nature of the places in which they are located.

The names for location scopes are based on a composite schema, named “lbs:”, that follows the generic syntax defined for Uniform Resource Identifiers (URI) [12]. The first component of the name, after the schema identifier “lbs:”, indicates a Location-Based Services (LBS) server (described in Sect. 3.4), and the second a location context managed by that server, as in the following example: “lbs:lbs.lancs.ac.uk/engineering”.

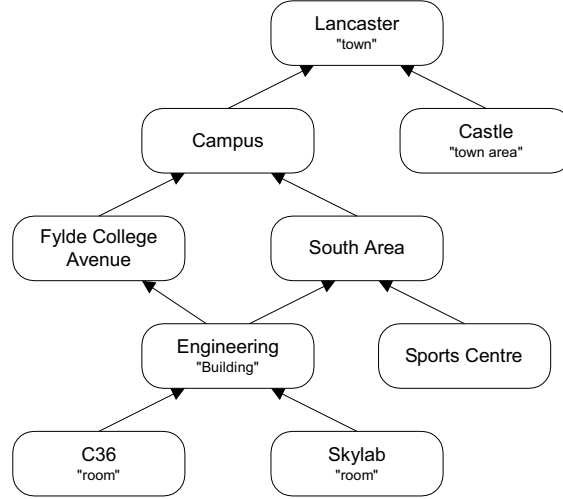


Fig. 1. An example of a location domain model

This naming approach for location contexts is based on two key decisions. The first has been to embed the name of the LBS server that manages the context. This speeds up the service location process by eliminating the extra level of indirection that would otherwise be required to obtain, given a location context name, the respective LBS server. The other key decision has been not to associate names with the hierarchy of location contexts, i.e. a location context is not necessarily a naming context for its contained locations. We expect the location hierarchy to be flexible and to support multiple views of the location space. Forcing the association between the organisation of locations and the organisation of naming would lead to a rigid and formal location hierarchy that would become a limiting factor for the dynamic of the context space. On the other hand, in some cases, simplicity may be more important than flexibility, as in the case of well structured places, such as a building. Whenever that is considered the best approach hierarchical names may also be used, e.g. “lbs:lbs.lancs.ac.uk/engineering/C36”.

3.3 Mapping Between Networks and Location Contexts

Location contexts are not bound to any specific network, but each participating network must be mapped onto a location context. Such mapping allows a device, without any *a priori* information, to determine its corresponding location context as a function of its point of attachment to the network. A mapping of a network portion to a location context means that the area of physical coverage of that network is contained within the area corresponding to the location context. Multiple networks may be mapped to the same location context representing the physical overlapping of the respective coverage areas.

Even though we assume that this mapping will be available in any participating network, we do not define any specific mechanisms for supporting it. The most appropriate technique must be chosen for each specific case, considering factors such as the specific network environment or the desired granularity. Beaconing techniques provide an adequate solution for simple network infrastructures, especially cellular networks, but may be limited for more complex network infrastructures, in which it is difficult to set up a beacon per link and to define the boundaries of multicast traffic. More elaborate and flexible methods, possibly based on a combination of new techniques, such as Administratively Scoped IP Multicast [13] or GPS based routing [10], are essential before this mapping can be widely supported in multiple network environments.

Alternative forms of determining location, such as GPS, may also be used either as a substitute for the lack of information from the network or as a way of improving its granularity. However, the need to map the location information obtained from those sources into valid location contexts implies that the system must have some information about the location contexts available for its area of operation.

3.4 LBS Servers

The architecture of our system is based on LBS servers. The primary function of an LBS server is to maintain a dynamic repository of service registrations and satisfy requests upon that information. Service registrations and requests are done on a per location basis and each server is responsible for maintaining the registrations corresponding to a number of locations. Each service registration is identified by a “service:” Uniform Resource Locator (URL)[14], and may include attributes for describing the characteristics of the service and aiding service selection. Registrations are only maintained for a certain period of time, unless they are refreshed. This soft-state approach is similar to the one used in SLP, but more flexible refresh intervals are used.

The other function of LBS servers is to support the hierarchy of location contexts. This involves maintaining information about the location contexts and their relationships. Information about each location context typically includes service location policies and attributes describing the location context, e.g. its type and geographical coverage. Each location also maintains a set of links to other locations in which it is contained. The next section analyses the effect of these associations in terms of service location.

The existence of relationships between location contexts maintained by different servers provides the basis for the creation of LBS federations, such as the one represented in Fig. 2. A federation allows servers to share their respective service offer spaces, and thus form a larger location space, but the loose integration between location contexts allows servers to keep their autonomy and set their own policies. This cooperative and distributed process of creating relationships between location contexts is the basis for supporting larger location models.

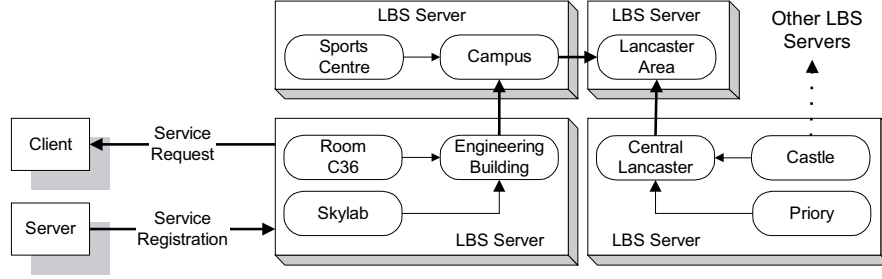


Fig. 2. A federation of LBS servers

3.5 Service Location

The overall process of service location can be described as follows. From its point of attachment to the network, and possibly with the collaboration of other locating technologies, a device determines its current location context. Since the name of the location context includes a reference to the respective LBS server, applications on the device (clients or servers) may start to interact with the appropriate server. This interaction is based on the exchange of protocol messages that are basically a subset and an adaptation of the messages used by SLP. Servers issue service registration messages, indicating the characteristics of their service and the location context at which it should be registered. Clients issue service request messages indicating the location context, the type of service required, and the attribute based criteria that the service must satisfy.

The Semantics of Containment. Location contexts are more than mere service aggregations, as the existence of relationships between locations has strong implications in the overall process of locating services. When a location context is contained within another, then the services registered in the containing location are also available in the contained location, subject only to systems policies. For example, a service registered in a “building” location should also be available to devices on the enclosed contexts, e.g. “room” locations. In contrast, a client searching for services in a “building” location should not receive information about services that are registered with “room” scopes. Supporting this cascaded service location means that an LBS server may have to forward a request to other LBS servers that maintain hierarchically superior location contexts before it can answer a service request.

Proximity. Another important aspect of service location is how multiple notions of proximity are supported by the system. Since there is no notion of service position, other than registration in a given location context, the basic level of proximity that a device may use to find services is the smallest location context in which it may assume to be.

Enlarging this proximity range, i.e. searching for services in a wider area, is achieved by changing the scope indicated in the service request to a higher location context. As a result, only the services that have been registered for that wider scope will be returned. The advantage of this approach is supporting a scalable notion of proximity, in which the range of proximate selection may be expanded without an exponential increase in the number of returned services. From an architectural point of view, this is substantially different from discovering the services that are located within a certain range, but it is consistent with our concept of service location in which the objective is to select services whose scope includes our current location context.

3.6 Implementation

We are in the process of implementing the various elements of this architecture. We have implemented an LBS server supporting multiple location contexts, contained relationships, and service requests. On the client side, we have a general functional model for the interaction of applications with the framework. The core of this interaction is an adapted version of the JAVA Application Program Interface (API) for SLP described in [15]. This API, modified to reflect the specificity of location-based services, provides an object oriented interface to the functionality of the framework and is designed to allow implementations to offer just the feature set they need. The use of this API greatly simplifies the development of applications and also allows common functionality, e.g. determining the current location context, to be shared by several applications. Also associated with this API, there is a file format for serialised service registrations, allowing legacy services to be registered and registration databases to be exchanged. All these implementation components have been used for building the case study described in the next section.

4 Case Study: A Location-Sensitive Tourist Guide

This section describes the application of our architectural framework to the development of a location-sensitive tourist guide. This case study is based on the requirements and infrastructure of the Guide system [1], currently under development at Lancaster University.

The Guide is a context-sensitive tourist guide for visitors to the city of Lancaster. As visitors, equipped with portable Guide units, move around the city, the system displays the information that is relevant for their current location. Examples of location-dependent information provided by the Guide are the local attractions, a map of the area, and a list of events. The system is based on a set of wireless cells, placed in key locations in town, that send beacons containing a reference to the information associated with that area.

4.1 Technical Description

The realisation of this case study involved four basic activities: the definition of the location model; the realisation of the supporting architecture; the creation of the location-based services; and the development of a location-dependent application that uses them to access information.

Location Model. The location model for the Guide case study is diagrammatically represented in Fig. 3.

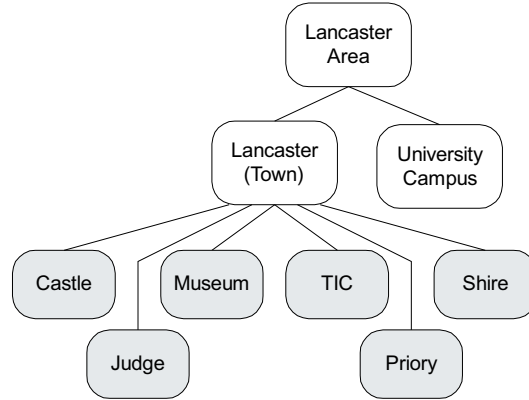


Fig. 3. The location model for the GUIDE case study

The model was designed to provide the maximum location granularity over the wireless cells of the Guide networking infrastructure and to take advantage of the respective beaconing system. This was achieved by a one to one mapping between each cell and a location context, originating the set of location contexts represented in grey in Fig. 3. In order to better explore the possibilities of location-based services, a simple hierarchy was created with the introduction of some additional location contexts in which the basic locations were included.

System Architecture. The architecture used for this case study is represented in Fig. 4, and is based on a single LBS server that manages all the location contexts of the model. All available services are registered here at the appropriate location contexts.

Since the association between services and locations is only dependent on the location context indicated in the service registration, and not on the position of servers, we used a single information server to support all the services created for this case study. Services were modelled as Extensible Markup Language (XML) documents available through the Hypertext Transfer Protocol (HTTP). Appropriate Document Type Definitions (DTD) were created for each type of service,

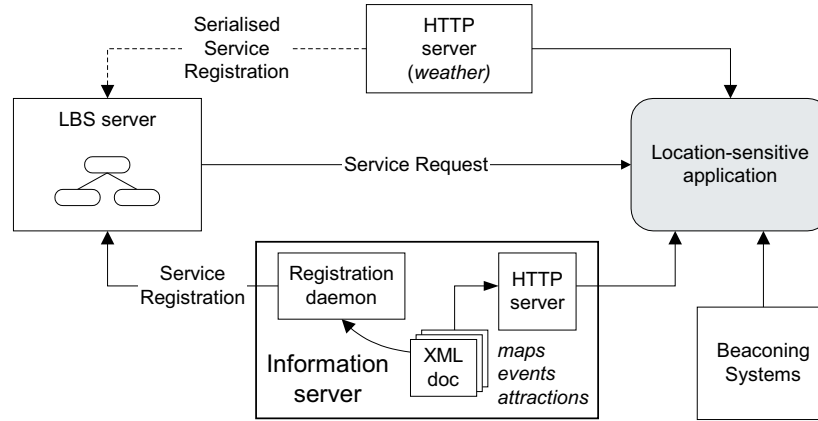


Fig. 4. The architecture of the location-sensitive tourist guide

but they all included data fields indicating the location context on which the service should be registered and its registration attributes. A registration daemon reads the XML documents and uses that information to register the respective services in the appropriate location contexts. The weather service was obtained from an existing server on the Internet, and its registration was based on a serialised service registration.

Location-Based Services. We have created a number of location-based services for satisfying the information needs of the tourist guide. More specifically, we have created services with information about local attractions, events in the area, maps, and weather forecast. The SLP service type definition [14] was used to distinguish between the several types of services, with an abstract service type being defined for each of the information services, i.e. map, weather, attractions and events. For example, the map service was identified as being of type *map:http*, being *map* the abstract type, which defines the nature of the information, and *http* the concrete type, which defines the network protocol by which it can be obtained.

Depending on the nature of services, different strategies were used in the distribution of services among location contexts. A service of type map was created for each location context. The map service registered at the Lancaster location provides a general town map, while the services in smaller locations provide more detailed maps of those areas. Several event services were also created for the various location contexts. The services in smaller locations offer more specific information about events that take place in the respective area, while the service for the town only lists the most important events, i.e. those whose scope surpasses the town area in which they occur. A similar approach was used with the attraction service, with the town wide service providing information about

the major attractions in town. The weather service, given its granularity, was only available at the Lancaster area scope.

The use of a location hierarchy supports some interesting options in the association of information with location. For example, town-wide information may easily be associated with the town location context, thus avoiding the need to replicate it on each cell as it was the case with the original Guide model. Another interesting feature is the possibility of selecting the most appropriate proximity level for obtaining information. For example, tourists without much time to visit the town may only be interested in the main town attractions. They may prefer to use the town-wide attractions service instead of a service specific to a smaller area, even if it is their own current location. In these cases, the ability to specify the proximity range becomes a form of indicating the desired information detail, which may be an important feature for supporting such different contexts of information access as walking or driving.

The Application. The visible component of the system is the location-sensitive application, running on a portable unit, that visitors can use to access the relevant information to their current location. It determines its location context from the GUIDE beacons it receives and from a simple mapping function that translates that information into valid location contexts. This information is then used to find the services it needs, obtain the required information from those servers and, if requested, display the information to the user. We have used basically the same interface of the Guide system, but the functionality associated with some buttons was slightly changed to better reflect the new underlying architecture and its possibilities. The interaction with the LBS server is done through the API created for this purpose, thus keeping to a minimum the effort of implementing the application.

4.2 Analysis

The realisation of this case study has allowed us to evaluate several aspects about the applicability of our framework and the LBS concept in general. Despite being a fairly simple example, it has been extremely useful in highlighting a number of open issues and potential difficulties. One of those issues is how to define and support the information space of an application, i.e. the information it requires for its normal operation, based only on service location. If all that the application needs is any server of a given type, then the problem is trivial. However, in more elaborate cases, such as a tourist guide, creating the information space of an application may imply interacting with a large number of servers and may involve many more aspects, such as the characteristics of the services, user preferences in service selection or service content.

In general, our approach appears to provide a useful framework for the development of location-dependent systems. Furthermore, the abstractions introduced by the use of location-based services have provided the basis for transforming an essentially self-contained system into an open system with a much larger

potential for evolution. New sources of information, even if not explicitly created for this purpose, may be added by simply registering new services. New types of information may be supported with the creation of new service types. Other networks, wireless or not, may be used for accessing the same services, depending only on the existence of a mapping to location contexts. The same information infrastructure may easily be used as the basis for new unpredicted applications. All these extensions are possible, and simple, because the abstractions on which the system design is based avoid hidden dependencies between its different components and provide a clear separation of concerns between its main functions such as information provision, location modelling, service location and application development.

5 Future Work and Conclusions

5.1 Future Work

Our work on location-based services is an ongoing project. We plan to extend our framework in order to address some of the existing issues, and we plan to develop further case studies. We are currently preparing a new, more ambitious, case study in the context of the AROUND project. This project, which has recently started at University of Minho, aims at providing transport related information, and will have a stronger focus on exploring the possibilities introduced by the use of location-based services. The prototype will have a larger number of services and service types, and will be based on multiple networks, including the Global System for Mobile communication (GSM). With the realisation of multiple case studies, with different characteristics in terms of applications and technological environments, we hope to evaluate the effect that several factors may have on the applicability of the LBS concept. The realisation of multiple case studies is also essential in demonstrating that the proposed framework can serve as a unified approach to several different scenarios, as this is one of its main advantages in comparison with existing approaches.

Security. We are aware that security is a crucial component of our framework. A comprehensive solution will require a careful analysis of the specific security requirements of location-based services and a clearer definition of the system administrative model. Our emphasis, so far, has been on the definition of such model, rather than on the introduction of ad hoc security mechanisms.

Hybrid Models. There are many cases in which plain service location may be inefficient for building the information space of an application. We will be looking for new solutions that, while still based on services and service location, may also combine some functionality commonly associated with other technologies, such as directory services and information retrieval. Depending on the nature of the services, prefetching and caching of service information may also be considered, not only as a way of optimising performance, but also as a way of supporting weakly connected operation.

Networking Environments. The emergence of new wireless network technologies based on short range connectivity, e.g. bluetooth [16], and the increasing number of devices equipped with network connections, possibly even more than one, represent a vast potential for applying the concept of location-based services to smaller location scales. Exploring the use of our framework in these highly networked environments is another objective of our future research.

5.2 Conclusions

There is a growing interest in systems that support associations between information and location, but there is a lack of generic approaches on which to base their development. As a consequence, developing such systems becomes a technically complex task, leading to partial solutions that are only valid for very specific applications and technological environments. In this paper, we have proposed a flexible concept of location-based services as a paradigm for supporting the use of localised information and described an architectural framework for enabling such service model in the Internet environment. The application of the framework has been demonstrated through the development of a location-sensitive tourist guide, which also served for evaluating the approach and for highlighting some of the issues involved in its use. The results of the case study have shown that the concept of location-based services can be used to develop location-dependent systems. Moreover, the characteristics of the resulting architecture, in terms of its openness and generality, suggest that location-based services can effectively be used as a more generic approach to support localised information.

Acknowledgements

To the GUIDE team, and especially to Keith Mitchell and Matthias Franz, for their collaboration in the preparation of this case study. To Adrian Friday for his comments on a draft version of this paper. To the anonymous reviewers for their attentive reading and valuable comments. This work was carried out as part of the PRAXIS funded AROUND project (PRAXIS/P/EEI/14267/1998) and supported by grant PRAXIS XXI/BD/13853/97.

References

1. Davies, N.,Cheverst, K.,Mitchell, K.,Friday, A.: Caches in the Air: Disseminating Tourist Information in the Guide System. Second IEEE Workshop on Mobile Computer Systems and Applications, New Orleans, Louisiana (1999)
2. Imielinski, T.,Viswanathan, S.: Adaptive Wireless Information Systems. SIG in Data Base Systems Conference, Tokyo, Japan (1994)
3. Hodes, T. D.,Katz, R.,Servan-Schreiber, E.,Rowe, L.: Composable Ad hoc Mobile Services for Universal Interaction. 3rd ACM/IEEE International Conference on Mobile Computing and Networking - MOBICOM97, Budapest, Hungary (1997)

4. Jini website, available as "<http://www.sun.com/jini>".
5. HAVi website, available as "<http://www.havi.org>".
6. Schilit, W.: A System for Context-Aware Mobile Computing. Columbia University, PhD Thesis (1995)
7. Brown, P.: Triggering Information by Context. Personal Technologies Vol. 2 1 (1998) 1-9
8. Guttman, E., Perkins, C., Veizades, J., Day, M.: Service Location Protocol, Version 2. RFC 2608 (1999)
9. Rosenberg, J., Schulzrinne, H., Suter, B.: Wide Area Network Service Location. Work in progress from the Service Location Working Group of the IETF, Internet-Draft draft-ietf-srvloc-warsv-01.txt (1997)
10. Imielinski, T., Navas, J.: GPS-Based Addressing and Routing, RFC 2009 (1996)
11. Leonhardt, U.: Supporting Location-Awareness in Open Distributed Systems. Imperial College of Science, Technology and Medicine, University of London, PhD Thesis (1998)
12. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396 (1998)
13. Meyer, D.: Administratively Scoped IP Multicast. RFC 2365 (1998)
14. Guttman, E., Perkins, C., Kempf, J.: Service Templates and Service: Schemes. RFC 2609 (1999)
15. Kempf, J., Guttman, E.: An API for Service Location. RFC 2614 (1999)
16. Bluetooth website, available as "<http://www.bluetooth.com>".